Service-Oriented Architecture for Privacy-Preserving Data Mashup

Thomas Trojer[a] Benjamin C. M. Fung[b]
Patrick C. K. Hung[c]

[a]Quality Engineering, Institute of Computer Science, University of Innsbruck, Austria

[b]Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada

[c]University of Ontario Institute of Technology, Oshawa, ON, Canada

July 9, 2009

Motivation

A lot of effort has been put into research of data-privacy preserving methods. We're working on applying such methods in real usecases and the design of appropriate architectural approaches and communication protocols.

Outline

- Introduction
  - Problem Statement
  - Related Work
- Privacy-preserving Data Mashup
  - Requirements
  - Multi-party anonymization
- Proposed Architecture
  - Components
  - Communication protocol
- Conclusion

What are data mashups?

- Web technology integrating knowledge from different sources
- Goal of obtaining new or more sophisticated knowledge from results
- Data mashups deal with integrated data from multiple data providers
  (We are currently assuming homogenously structured data tables)

Problem Statement

- Collaborative project with Nordax Finans AB, a provider of unsecured loans in Sweden.
- A generalization of the problem is described as
  - Loan company A and bank B observe different sets of attributes of individuals, e.g. $T_A(SSN, Age, Balance)$ and $T_B(SSN, Job, Salary)$
  - Individuals are commonly identified by their SSN (social security number) in both record tables given
  - Goal of implementing a data-mashup application to better support general loan or credit limit approval.

Privacy issues in data mashups

- Simply joining tables T_A and T_B would reveal sensitive information to the opponent party
- Single-party generalized and integrated data tables hinder support for data mining and classification analysis on top of the joined data.
- Pre-anonymized data tables joined would possibly still let a participating party identify certain records
  . . . , but this depends on an appropriate anonymization procedure

Classification analysis is strongly bound to patterns obtained from the datasets, therefore we have to preserve typical structures related to our data and simultaneously considering certain privacy concerns.

Related Work

- Information integration mostly database research, literature typically assumes information to be freely shared.
  - G. Wiederhold. Intelligent integration of information. 1993 ACM SIGMOD.
  - R. Agrawal, A. Evfimievski, and R. Srikant. Information sharing across private databases. 2003 ACM SIGMOD.
- Secure multiparty computation allows sharing of computed results (e.g. a classifier) but prohibits sharing of data.
  - C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin, and M. Y. Zhu. Tools for privacy preserving data mining. SIGKDD Explorations.
  - W. Du and Z. Zhan. Building decision tree classifier on private data. Workshop on Privacy, Security, and Data Mining at the 2002 IEEE ICDM.

Related Work (cont.)

- Privacy-related notions like *k*-anonymity where used in several systems (e.g. Datafly system).
  - P. Samarati and L. Sweeney. Generalizing data to provide anonymity when disclosing information. 17th ACM PODS.
- Secure integration of data proposed via using cryptographic approaches, do not consider data mining tasks.
  - W. Jiang and C. Clifton. Privacy-preserving distributed k-anonymity. 19th Annual IFIP WG 11.3 Working Conference on Data and Applications Security.

Pre-requisites

- Requirements to fulfull,
  1. Satisfying the joint-anonymity requirement
  2. Maximizing classifyable data
  3. Each involved party doesn't learn more than what is given with a data table fulfilling (1)
- Anonymization protocol to extend for our multi-party purpose
  (in our case Top-down Specialization (TDS))
- Architectural approach to adopt

Requirements

1. Joint-anonymity requirement
   - Specific data attributes in combination identify individuals (Quasi Identifiers (QID))
   - Considering all QIDs of a data table for privacy protection
   - *k*-Anonymity, each value-combination of a QID is shared by at least *k* records of a data table *T*

Requirements (cont.)

2. Data for classification analysis
   - Generalized data shall stay as useful for classification as possible
   - Privacy goal requires us to mask information specific to individuals . . .
   - . . . Information requirement forces us to preserve certain structures for analysis

Generalization has to be performed in a “carefully” way!

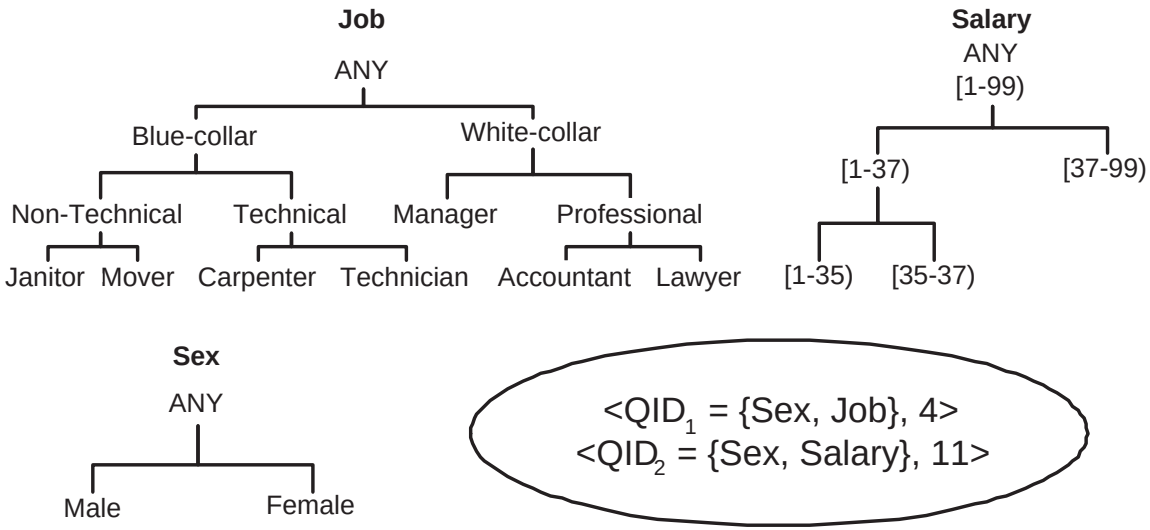
Requirements (cont.)

3. Multiple private data tables
   - Different owners of data won't share knowledge (The loan company and the bank are trusted to secure the privacy of their customers)
   - All parties have to communicate metadata of their data tables and the controlling of the generalization procedure

Multiple private data tables get translated in one integrated and privacy-preserved table, where no participant is able to obtain knowledge about individuals other than what was known to her/him before.

Multi-party anonymization

- Single data table specialization using TDS approach
  - Categorical attributes related to taxonomy tree
  - Numerical attributes related to arbitrary large value ranges
  - Values starting from most general (top) and get more specific (top-down) until a privacy violation would occur (in our usecase, understepping *k*)



Multi-party anonymization (cont.)

- Value of a specialization step
  - Information gain versus anonymity loss (Score)
  - Information gain is based on the difference in entropy before and after specialization
  - Anonymity loss is based on the average change in the number of values of a certain attribute before and after specialization

Multi-party anonymization (cont.)

$$InfoGain(v) = I(T[v]) - \sum_c \frac{|T[c]|}{|T[v]|} * I(T[c]),$$

where $T[v]$ is the set of records with an attribute (specialized to) v. $I(T[v])$ defines the entropy of $T[v]$ by,

$$I(T[v]) = -\sum_{cls} \frac{freq(T[v],cls)}{|T[v]|} * log_2 \frac{freq(T[v],cls)}{|T[v]|},$$

where cls is a specific class-value of attribute v and $freq(T[v], cls)$ represents the frequency of the value cls in the set of records $T[v]$.

Multi-party anonymization (cont.)

$$AnonyLoss(v) = avg(A(QID_j) - A_v(QID_j)),$$

where $avg(z)$ is a function calculating the average of values contained in any numeric set z and $A(QID_j)$, $A_v(QID_j)$ represent the count of any value qid_j of QID_j least frequent in the total set of records, before and after specializing v respectively. Note, $A(QID) > k$ has to hold throughout and is called anonymity.

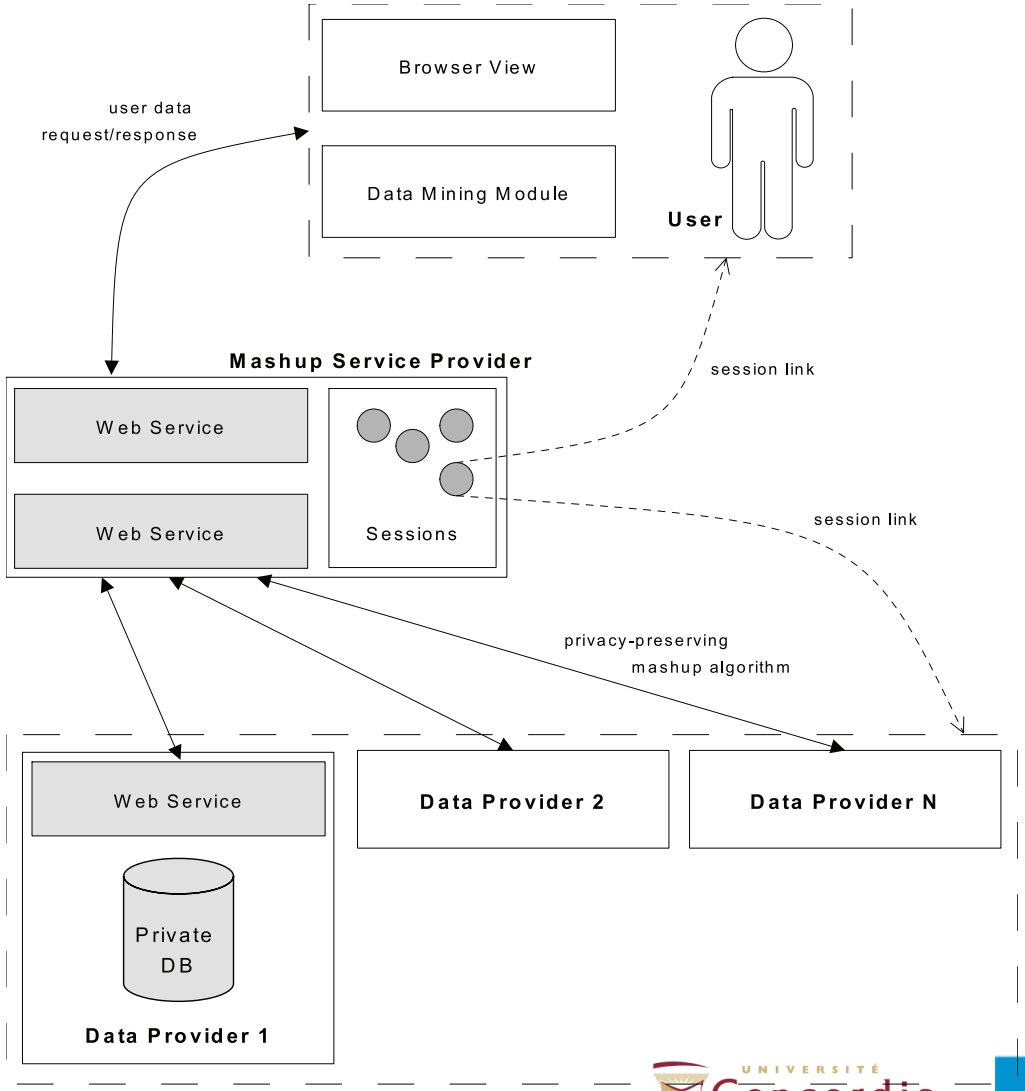
$$Score(v) = \frac{InfoGain(v)}{AnonyLoss(v)+1}$$

Components

Components defined by the mashup architecture

- Service client
  - Processing layer
    (e.g. data mining module)
  - Presentation layer
    (presentation specification, e.g. browser view)
- Mashup service provider
  - Session linking
  - Domain knowledge
    (which data providers and how to correlate them together?)
  - Security and data privacy responsibility
    (i.e. coordinating PPMashup)
- Data provider
  - Responsible for local private database
  - Data privacy responsibility
    (i.e. participating in PPMashup)

Components (cont.)



Phase I, Session establishment

- Establishing an operational context,
  - Step 1 Authentication and authorization of service client
  - Step 2 Identification and selection of contributing data providers
  - Step 3 Initialization of a common session context (i.e. linking the service client to the set of chosen data providers)
  - Step 4 Common requirements negotiation
    - Information requirements (SLA's, data pieces of interest)
    - Data privacy requirements (negotiated properties for instantiating a PPMashup run)

Phase I, Mashup request/response

- Mashup application request
  (implicitly given with session establishment)
- Mashup application response
  - Joined resulting data table
  - Data-privacy preserved according to PPMashup

Phase II, Privacy-preserving protocol

- Initiation of PPMashup by mashup provider
  - Mashup provider only accesses final integrated table
  - Computational burden is on single data providers
- Protocol coordination
  - Linking data providers to one common session context
  - Evaluating *Score*(x) values communicated by using the Secure Multiparty Maximum protocol
  - Notification of all participants on the winner candidate and guidance on the specification

Phase II, Multi-party algorithm

initialize T_g to include one record containing top most values;
initialize $\cup Cut_i$ to include only top most values;

while(there is some candidate in $\cup Cut_i$ **)** {
    find the local candidate x of highest $Score(x)$;
    communicate $Score(x)$ with MP to find the winner;

    if(the winner w is local (notified by MP) **)** {
        specialize w on T_g;
        forward to MP to instruct other DPs to specialize w;
    } **else** {
        wait for the instruction from MP;
        specialize w on T_g using the instruction;
    }

    replace w with $child(w)$ in the local copy of $\cup Cut_i$;
    update $Score(x)$, the beneficial/valid status for candidates x in $\cup Cut_i$;
}

output T_g and $\cup Cut_i$ to MP;

Conclusion

- Loose coupling in this specific architectural domain
- Mashup coordinator works as kind of mediator
- Currently only datastructures representing relations (n-tuples)
- Score function is extensible/exchangable
  (e.g. making use of additional privacy measures, like *l*-diversity)
- Simple, yet powerful architecture
  - Mashup provider/Data providers could be used hierarchically
    (anonymized data could be cached at mashup provider side)
  - Mashup provider makes use of data model specification presented in WS description
    (validation of QIDs or (semantic) matching could be applied here)

Thanks for listening!

Questions and comments are very appreciated